D 16

```
FFFFFFFFFFFFFFF        111              111            AAAAAAAAA
FFFFFFFFFFFFFFF        111              111            AAAAAAAAA
FFFFFFFFFFFFFFF        111              111            AAAAAAAAA
FFF                 111111           111111         AAA         AAA
FFF                 111111           111111         AAA         AAA
FFF                 111111           111111         AAA         AAA
FFF                    111              111         AAA         AAA
FFF                    111              111         AAA         AAA
FFF                    111              111         AAA         AAA
FFFFFFFFFFFF           111              111         AAA         AAA
FFFFFFFFFFFF           111              111         AAA         AAA
FFFFFFFFFFFF           111              111         AAA         AAA
FFF                    111              111         AAAAAAAAAAAAAAA
FFF                    111              111         AAAAAAAAAAAAAAA
FFF                    111              111         AAAAAAAAAAAAAAA
FFF                    111              111         AAA         AAA
FFF                    111              111         AAA         AAA
FFF                    111              111         AAA         AAA
FFF                 111111111        111111111      AAA         AAA
FFF                 111111111        111111111      AAA         AAA
FFF                 111111111        111111111      AAA         AAA
```

```
                                                       M 16
**FILE**ID**DELETE








DDDDDDDD    EEEEEEEEEE  LL          EEEEEEEEEE  TTTTTTTTTT  EEEEEEEEEE
DDDDDDDD    EEEEEEEEEE  LL          EEEEEEEEEE  TTTTTTTTTT  EEEEEEEEEE
DD      DD  EE          LL          EE              TT      EE
DD      DD  EE          LL          EE              TT      EE
DD      DD  EE          LL          EE              TT      EE
DD      DD  EE          LL          EE              TT      EE
DD      DD  EEEEEEEE    LL          EEEEEEEE        TT      EEEEEEEE
DD      DD  EEEEEEEE    LL          EEEEEEEE        TT      EEEEEEEE
DD      DD  EE          LL          EE              TT      EE
DD      DD  EE          LL          EE              TT      EE
DD      DD  EE          LL          EE              TT      EE                ....
DD      DD  EE          LL          EE              TT      EE                ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  EEEEEEEEEE      TT      EEEEEEEEEE      ....
DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  EEEEEEEEEE      TT      EEEEEEEEEE      ....


LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 MODULE DELETE (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 ! FACILITY:  F11ACP Structure Level 1
1 !
1 ! ABSTRACT:
1 !
1 !       This routine performs the DELETE function.
1 !
1 ! ENVIRONMENT:
1 !
1 !       STARLET operating system, including privileged system services
1 !       and internal exec routines.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  1-Apr-1977
1 !
1 ! MODIFIED BY:
1 !
1 !       A0101   ACG0116         Andrew C. Goldstein,    16-Jan-1980  16:47
1 !               Swallow errors reading the file header
1 !
1 !       A0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  20:02
1 !       Previous revision history moved to F11A.REV
1 !**
1
```

```
DELE
V04-
;
; Si
; Ru
; El
; Li
; Le
; Me
; Co
```

```
:    58         0058  1
:    59         0059  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
:    60         0060  1 REQUIRE 'SRC$:FCPDEF.B32';
:    61         0375  1
:    62         0376  1
:    63         0377  1 FORWARD ROUTINE
:    64         0378  1         DELETE,                         ! main delete function
:    65         0379  1         MARK_DELETE     : NOVALUE,      ! mark file for delete
:    66         0380  1         MARKDEL_FCB,                    ! mark FCB of file for delete
:    67         0381  1         DELETE_HANDLER;                 ! condition handler for delete function
:    68         0382  1
:    69         0383  1 OWN
:    70         0384  1         SAVE_STATUS;                    ! place to save user status for handler
```

```
;     72          0385  1 GLOBAL ROUTINE DELETE =
;     73          0386  1
;     74          0387  1 !++
;     75          0388  1 !
;     76          0389  1 ! FUNCTIONAL DESCRIPTION:
;     77          0390  1 !
;     78          0391  1 !       This routine performs the remove and mark for delete functions.
;     79          0392  1 !
;     80          0393  1 ! CALLING SEQUENCE:
;     81          0394  1 !       DELETE ()
;     82          0395  1 !
;     83          0396  1 ! INPUT PARAMETERS:
;     84          0397  1 !       NONE
;     85          0398  1 !
;     86          0399  1 ! IMPLICIT INPUTS:
;     87          0400  1 !       IO_PACKET: I/O packet in process
;     88          0401  1 !
;     89          0402  1 ! OUTPUT PARAMETERS:
;     90          0403  1 !       PRIMARY_FCB: FCB of file
;     91          0404  1 !
;     92          0405  1 ! IMPLICIT OUTPUTS:
;     93          0406  1 !       NONE
;     94          0407  1 !
;     95          0408  1 ! ROUTINE VALUE:
;     96          0409  1 !       1
;     97          0410  1 !
;     98          0411  1 ! SIDE EFFECTS:
;     99          0412  1 !       directory entry removed
;    100          0413  1 !       file marked for delete or deleted
;    101          0414  1 !
;    102          0415  1 !--
;    103          0416  1
;    104          0417  2 BEGIN
;    105          0418  2
;    106          0419  2 LOCAL
;    107          0420  2         ABD             : REF BBLOCKVECTOR [,ABD$C_LENGTH],
;    108          0421  2                                         ! buffer descriptors
;    109          0422  2         FIB             : REF BBLOCK;   ! FIB
;    110          0423  2
;    111          0424  2 EXTERNAL
;    112          0425  2         CLEANUP_FLAGS   : BITVECTOR,    ! cleanup action flags
;    113          0426  2         IO_PACKET       : REF BBLOCK,   ! I/O packet in process
;    114          0427  2         CURRENT_WINDOW  : REF BBLOCK,   ! window of open file
;    115          0428  2         PRIMARY_FCB     : REF BBLOCK,   ! FCB of file
;    116          0429  2         CURRENT_VCB     : REF BBLOCK;   ! VCB of volume
;    117          0430  2
;    118          0431  2 EXTERNAL ROUTINE
;    119          0432  2         GET_FIB,                        ! get FIB of request
;    120          0433  2         FIND,                           ! find name in directory
;    121          0434  2         SEARCH_FCB;                     ! search FCB list
;    122          0435  2
;    123          0436  2
;    124          0437  2 ! First find the buffer descriptor, FIB, FCB, etc. then remove the
;    125          0438  2 ! directory entry.
;    126          0439  2 !
;    127          0440  2
;    128          0441  2                                         ! pointer to buffer descriptors
;   129          0442  2  ABD = .BBLOCK [.IO_PACKET[IRP$L_SVAPTE], AIB$L_DESCRIPT];
;   130          0443  2  FIB = GET_FIB (.ABD);
;   131          0444  2
;   132          0445  2  IF .CURRENT_VCB[VCB$V_NOALLOC]
;   133          0446  2  THEN ERR_EXIT (SS$_WRITLCK);
;   134          0447  2
;   135          0448  2  ! If a directory ID is present, do a directory search first and remove
;   136          0449  2  ! the directory entry.
;   137          0450  2  !
;   138          0451  2
;   139          0452  2  IF .FIB[FIB$W_DID_NUM] NEQ 0
;   140          0453  2  THEN FIND (.ABD, .FIB, 1);
;   141          0454  2
;   142          0455  2  ! If there is a file open on the channel, check the file ID returned by the
;   143          0456  2  ! FIND against that of the open file. If they do not match, treat the file
;   144          0457  2  ! as if it were not open.
;   145          0458  2  !
;   146          0459  2
;   147          0460  2  IF .PRIMARY_FCB NEQ 0
;   148          0461  2  THEN
;   149          0462  3      BEGIN
;   150          0463  3      IF .PRIMARY_FCB[FCB$W_FID_NUM] NEQ .FIB[FIB$W_FID_NUM]
;   151          0464  3      OR .PRIMARY_FCB[FCB$W_FID_SEQ] NEQ .FIB[FIB$W_FID_SEQ]
;   152          0465  3      THEN CURRENT_WINDOW = 0;
;   153          0466  2      END;
;   154          0467  2
;   155          0468  2  PRIMARY_FCB = SEARCH_FCB (FIB[FIB$W_FID]);
;   156          0469  2
;   157          0470  2  ! Now actually mark the file for delete if requested.
;   158          0471  2  !
;   159          0472  2
;   160          0473  2  IF .BBLOCK [IO_PACKET[IRP$W_FUNC], IO$V_DELETE]
;   161          0474  2  THEN MARK_DELETE (.FIB);
;   162          0475  2
;   163          0476  2  RETURN 1;
;   164          0477  2
;   165          0478  1  END;                                        ! end of routine DELETE


                                                                 .TITLE  DELETE
                                                                 .IDENT  \V04-000\

                                                                 .PSECT  $LOCKEDD1$,NOEXE,2

                                                   00000 SAVE_STATUS:
                                                                 .BLKB   4

                                                                 .EXTRN  CLEANUP_FLAGS, IO_PACKET
                                                                 .EXTRN  CURRENT_WINDOW, PRIMARY_FCB
                                                                 .EXTRN  CURRENT_VCB, GET_FIB
                                                                 .EXTRN  FIND, SEARCH_FCB

                                                                 .PSECT  $CODE$,NOWRT,2

                                           000C 00000            .ENTRY  DELETE, Save R2,R3                          ; 0385
                           50      0000G  CF  D0 00002            MOVL    IO_PACKET, R0                               ; 0442
                           53         2C  B0  D0 00007            MOVL    @44(R0), ABD                                ;
```

```
                                    53  DD 0000B         PUSHL   ABD                              ; 0443
               0000G  CF            01  FB 0000D         CALLS   #1, GET_FIB                      ;
                      52            50  D0 00012         MOVL    R0, FIB                          ;
                      50     0000G  CF  D0 00015         MOVL    CURRENT_VCB, R0                  ; 0445
        05       0B   A0            04  E1 0001A         BBC     #4, 11(R0), 1$                   ;
                             025C   8F  BF 0001F         CHMU    #604                             ; 0446
                                        04 00023         RET                                      ;
                               0A   A2  B5 00024 1$:     TSTW    10(FIB)                          ; 0452
                                    0B  13 00027         BEQL    2$                               ;
                                    01  DD 00029         PUSHL   #1                               ; 0453
                                    52  DD 0002B         PUSHL   FIB                              ;
                                    53  DD 0002D         PUSHL   ABD                              ;
               0000G  CF            03  FB 0002F         CALLS   #3, FIND                         ;
                      50     0000G  CF  D0 00034 2$:     MOVL    PRIMARY_FCB, R0                  ; 0460
                                    12  13 00039         BEQL    4$                               ;
                 04   A2       24   A0  B1 0003B         CMPW    36(R0), 4(FIB)                   ; 0463
                                    07  12 00040         BNEQ    3$                               ;
                 06   A2       26   A0  B1 00042         CMPW    38(R0), 6(FIB)                   ; 0464
                                    04  13 00047         BEQL    4$                               ;
                             0000G  CF  D4 00049 3$:     CLRL    CURRENT_WINDOW                   ; 0465
                               04   A2  9F 0004D 4$:     PUSHAB  4(FIB)                           ; 0468
               0000G  CF            01  FB 00050         CALLS   #1, SEARCH_FCB                   ;
               0000G  CF            50  D0 00055         MOVL    R0, PRIMARY_FCB                  ;
                      50     0000G  CF  D0 0005A         MOVL    IO_PACKET, R0                    ; 0473
                      07       21   A0  E9 0005F         BLBC    33(R0), 5$                       ;
                                    52  DD 00063         PUSHL   FIB                              ; 0474
               0000V  CF            01  FB 00065         CALLS   #1, MARK_DELETE                  ;
                      50            01  D0 0006A 5$:     MOVL    #1, R0                           ; 0476
                                        04 0006D         RET                                      ; 0478

; Routine Size:  110 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 GLOBAL ROUTINE MARK_DELETE (FIB) : NOVALUE =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine marks the indicated file for delete and deletes it
1 !       if it is not accessed.
1 !
1 ! CALLING SEQUENCE:
1 !       MARK_DELETE (ARG1)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of FIB
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       file marked for delete or deleted
1 !
1 !--
1
2 BEGIN
2
2 BUILTIN
2       FP;
2
2 MAP
2       FIB             : REF BBLOCK;   ! FIB
2
2 LOCAL
2       FCB             : REF BBLOCK,   ! FCB of file
2       HEADER          : REF BBLOCK;   ! file header
2
2 OWN
2       DELETELOGENB    : INITIAL (0);  ! delete logger enable
2
2 EXTERNAL
2       USER_STATUS,                    ! status to return to user
2       CLEANUP_FLAGS   : BITVECTOR,    ! cleanup action flags
2       CURRENT_WINDOW  : REF BBLOCK;   ! window of open file
2
2 EXTERNAL ROUTINE
2       SEARCH_FCB,                     ! search FCB list
2       READ_HEADER,                    ! read file header
2       LOG_DELETE,                     ! send message to delete logger
2       CHECK_PROTECT,                  ! check file protection
```

```
;   224         0536  2             MARK_DIRTY,                         ! mark buffer for write-back
;   225         0537  2             DELETE_FILE,                        ! delete the file
;   226         0538  2             CHECKSUM;                           ! checksum file header
;   227         0539  2
;   228         0540  2
;   229         0541  2     ! Check that the file is not a reserved file (FID less than 5).
;   230         0542  2     !
;   231         0543  2
;   232         0544  2     IF .FIB[FIB$W_FID_NUM] LSSU 5
;   233         0545  2     THEN ERR_EXIT (SS$_NOPRIV);
;   234         0546  2
;   235         0547  2     ! Find the FCB, if any, and then read the header. Reading the header is done
;   236         0548  2     ! under a condition handler that quietly exits with success if errors are
;   237         0549  2     ! encountered. Thus, deleting a bad file header succeeds quietly.
;   238         0550  2     !
;   239         0551  2
;   240         0552  2     FCB = SEARCH_FCB (FIB[FIB$W_FID]);
;   241         0553  2     SAVE_STATUS = .USER_STATUS;
;   242         0554  2     .FP = DELETE_HANDLER;
;   243         0555  2     HEADER = READ_HEADER (FIB[FIB$W_FID], .FCB);
;   244         0556  2     .FP = 0;
;   245         0557  2
;   246         0558  2     IF .DELETELOGENB THEN LOG_DELETE (.FIB, .HEADER);
;   247         0559  2
;   248         0560  2     ! Check file protection. Check if the file is write accessed by someone
;   249         0561  2     ! else and not the deleter.
;   250         0562  2     !
;   251         0563  2
;   252         0564  2     CHECK_PROTECT (DELETE_ACCESS, .HEADER, .FCB);
;   253         0565  3     IF (
;   254         0566  3         IF .FCB EQL 0
;   255         0567  3         THEN 0
;   256         0568  3         ELSE .FCB[FCB$W_WCNT] NEQ 0)
;   257         0569  3     AND (
;   258         0570  3         IF .CURRENT_WINDOW EQL 0
;   259         0571  3         THEN 1
;   260         0572  3         ELSE NOT .CURRENT_WINDOW[WCB$V_WRITE])
;   261         0573  2     THEN ERR_EXIT (SS$_ACCONFLICT);
;   262         0574  2
;   263         0575  2     CLEANUP_FLAGS[CLF_REENTER] = 0;         ! from now on deletion proceeds
;   264         0576  2
;   265         0577  2
;   266         0578  2     ! Mark the file for delete. If the file is not accessed, then proceed to
;   267         0579  2     ! actually delete it.
;   268         0580  2     !
;   269         0581  2
;   270         0582  2     HEADER[FH1$V_MARKDEL] = 1;
;   271         0583  2     MARK_DIRTY (.HEADER);
;   272         0584  2
;   273         0585  3     IF KERNEL_CALL (MARKDEL_FCB, .FCB)
;   274         0586  2     THEN DELETE_FILE (.FIB, .HEADER)
;   275         0587  2     ELSE CHECKSUM (.HEADER);
;   276         0588  2
;   277         0589  1     END;                                        ! end of routine MARK_DELETE
```

```
                                                       .PSECT  $LOCKEDD1$,NOEXE,2

                              00000000  00004 DELETELOGENB:
                                                       .LONG   0                                    ;

                                                       .EXTRN  USER_STATUS, READ_HEADER
                                                       .EXTRN  LOG_DELETE, CHECK_PROTECT
                                                       .EXTRN  MARK_DIRTY, DELETE_FILE
                                                       .EXTRN  CHECKSUM, SYS$CMKRNL

                                                       .PSECT  $CODE$,NOWRT,2

                                      000C 00000       .ENTRY  MARK_DELETE, Save R2,R3             ; 0479
                 50        04    AC    D0 00002        MOVL    FIB, R0                             ; 0544
                 05        04    A0    B1 00006        CMPW    4(R0), #5                           ;
                                 03    1E 0000A        BGEQU   1$                                  ;
                                 24    BF 0000C        CHMU    #36                                 ; 0545
                                       04 0000E        RET                                         ;
   52      04    AC              04    C1 0000F 1$:    ADDL3   #4, FIB, R2                         ; 0552
                                 52    DD 00014        PUSHL   R2                                  ;
         0000G   CF              01    FB 00016        CALLS   #1, SEARCH_FCB                      ;
                 53              50    D0 0001B        MOVL    R0, FCB                             ;
         0000'   CF      0000G   CF    D0 0001E        MOVL    USER_STATUS, SAVE_STATUS            ; 0553
                 6D      0000V   CF    9E 00025        MOVAB   DELETE_HANDLER, (FP)                ; 0554
                                 0C    BB 0002A        PUSHR   #^M<R2,R3>                          ; 0555
         0000G   CF              02    FB 0002C        CALLS   #2, READ_HEADER                     ;
                 52              50    D0 00031        MOVL    R0, HEADER                          ;
                                 6D    D4 00034        CLRL    (FP)                                ; 0556
                 0A      0000'   CF    E9 00036        BLBC    DELETELOGENB, 2$                    ; 0558
                                 52    DD 0003B        PUSHL   HEADER                              ;
                           04    AC    DD 0003D        PUSHL   FIB                                 ;
         0000G   CF              02    FB 00040        CALLS   #2, LOG_DELETE                      ;
                                 0C    BB 00045 2$:    PUSHR   #^M<R2,R3>                          ; 0564
                                 02    DD 00047        PUSHL   #2                                  ;
         0000G   CF              03    FB 00049        CALLS   #3, CHECK_PROTECT                   ;
                                 53    D5 0004E        TSTL    FCB                                 ; 0566
                                 16    13 00050        BEQL    4$                                  ;
                           1C    A3    B5 00052        TSTW    28(FCB)                             ; 0568
                                 11    13 00055        BEQL    4$                                  ;
                 50      0000G   CF    D0 00057        MOVL    CURRENT_WINDOW, R0                  ; 0570
                                 05    13 0005C        BEQL    3$                                  ;
   05      0B    A0              01    E0 0005E        BBS     #1, 11(R0), 4$                      ; 0572
                         0800    8F    BF 00063 3$:    CHMU    #2048                               ; 0573
                                       04 00067        RET                                         ;
         0000G   CF        80    8F    8A 00068 4$:    BICB2   #128, CLEANUP_FLAGS+2               ; 0575
            0D   A2        80    8F    88 0006E        BISB2   #128, 13(HEADER)                    ; 0582
                                 52    DD 00073        PUSHL   HEADER                              ; 0583
         0000G   CF              01    FB 00075        CALLS   #1, MARK_DIRTY                      ;
                                 53    DD 0007A        PUSHL   FCB                                 ; 0585
                                 01    DD 0007C        PUSHL   #1                                  ;
                                 5E    DD 0007E        PUSHL   SP                                  ;
                         0000V   CF    9F 00080        PUSHAB  MARKDEL_FCB                         ;
     00000000G   9F              04    FB 00084        CALLS   #4, @#SYS$CMKRNL                    ;
                 0B              50    E9 0008B        BLBC    R0, 5$                              ;
                                 52    DD 0008E        PUSHL   HEADER                              ; 0586
                           04    AC    DD 00090        PUSHL   FIB                                 ;
         0000G   CF              02    FB 00093        CALLS   #2, DELETE_FILE                     ;
```

```
                                              04 00098          RET                                   ;
                                          52  DD 00099 5$:      PUSHL   HEADER                        ; 0587
                      0000G  CF           01  FB 0009B          CALLS   #1, CHECKSUM                  ;
                                              04 000A0          RET                                   ; 0589

; Routine Size:  161 bytes,    Routine Base:  $CODE$ + 006E
;  279         0590  1  GLOBAL ROUTINE MARKDEL_FCB (FCB) =
;  280         0591  1
;  281         0592  1  !++
;  282         0593  1  !
;  283         0594  1  ! FUNCTIONAL DESCRIPTION:
;  284         0595  1  !
;  285         0596  1  !       This routine marks the FCB for the current file, if any, for delete.
;  286         0597  1  !       This routine must be executed in kernel mode.
;  287         0598  1  !
;  288         0599  1  ! CALLING SEQUENCE:
;  289         0600  1  !       MARKDEL_FCB (ARG1)
;  290         0601  1  !
;  291         0602  1  ! INPUT PARAMETERS:
;  292         0603  1  !       ARG1: address of FCB
;  293         0604  1  !
;  294         0605  1  ! IMPLICIT INPUTS:
;  295         0606  1  !       PRIMARY_FCB: FCB of file if any
;  296         0607  1  !       DIR_FCB: FCB of directory file if any
;  297         0608  1  !       CURRENT_VCB: VCB of volume
;  298         0609  1  !
;  299         0610  1  ! OUTPUT PARAMETERS:
;  300         0611  1  !       NONE
;  301         0612  1  !
;  302         0613  1  ! IMPLICIT OUTPUTS:
;  303         0614  1  !       NONE
;  304         0615  1  !
;  305         0616  1  ! ROUTINE VALUE:
;  306         0617  1  !       1 if FCB removed (i.e., if file is deletable)
;  307         0618  1  !       0 if delete is to be deferred
;  308         0619  1  !
;  309         0620  1  ! SIDE EFFECTS:
;  310         0621  1  !       FCB marked for delete, may be deallocated
;  311         0622  1  !
;  312         0623  1  !--
;  313         0624  1
;  314         0625  2  BEGIN
;  315         0626  2
;  316         0627  2  MAP
;  317         0628  2          FCB             : REF BBLOCK;   ! FCB arg
;  318         0629  2
;  319         0630  2  LOCAL
;  320         0631  2          DUMMY;                          ! dummy destination for REMQUE
;  321         0632  2
;  322         0633  2  EXTERNAL
;  323         0634  2          CURRENT_VCB     : REF BBLOCK,   ! VCB of volume
;  324         0635  2          PRIMARY_FCB     : REF BBLOCK,   ! FCB of current file
;  325         0636  2          DIR_FCB         : REF BBLOCK;   ! FCB of directory file
;  326         0637  2
;  327         0638  2  EXTERNAL ROUTINE
;  328         0639  2          DEL_EXTFCB,                     ! delete extension FCB's
;  329         0640  2          DEALLOCATE;                     ! deallocate dynamic memory
;  330         0641  2
;  331         0642  2
;  332         0643  2  ! If the FCB exists, we mark it for delete (causing the file to be deleted
;  333         0644  2  ! when the access count goes to 0). Make sure that the directory LRU bit
;  334         0645  2  ! in the FCB is off (crediting 1 to the LRU count if it was on). If the
;  335         0646  2  ! access count is zero, dump the FCB and its extensions.
;  336        0647  2 !
;  337        0648  2
;  338        0649  2 IF .FCB NEQ 0
;  339        0650  2 THEN
;  340        0651  3     BEGIN
;  341        0652  3     FCB[FCB$V_MARKDEL] = 1;
;  342        0653  3     IF TESTBITSC (FCB[FCB$V_DIR])
;  343        0654  3     THEN CURRENT_VCB[VCB$B_LRU_LIM] = .CURRENT_VCB[VCB$B_LRU_LIM] + 1;
;  344        0655  3     IF .FCB[FCB$W_ACNT] NEQ 0 THEN RETURN 0; ! file is accessed
;  345        0656  3     REMQUE (.FCB, DUMMY);
;  346        0657  3     DEL_EXTFCB (.FCB);
;  347        0658  3     DEALLOCATE (.FCB);
;  348        0659  3     IF .PRIMARY_FCB EQL .FCB THEN PRIMARY_FCB = 0;
;  349        0660  3     IF .DIR_FCB EQL .FCB THEN DIR_FCB = 0;
;  350        0661  2     END;
;  351        0662  2
;  352        0663  2 RETURN 1;                                  ! ok to delete file
;  353        0664  2
;  354        0665  1 END;                                       ! end of routine MARKDEL_FCB


                                                                   .EXTRN  DIR_FCB, DEL_EXTFCB
                                                                   .EXTRN  DEALLOCATE

                                         0000 00000                .ENTRY  MARKDEL_FCB, Save nothing        ; 0590
                    51          04   AC   D0 00002                 MOVL    FCB, R1                          ; 0649
                                     41   13 00006                 BEQL    3$
              22    A1               02   88 00008                 BISB2   #2, 34(R1)                       ; 0652
       08     22    A1               00   E5 0000C                 BBCC    #0, 34(R1), 1$                   ; 0653
                    50       0000G   CF   D0 00011                 MOVL    CURRENT_VCB, R0                  ; 0654
                                49   A0   96 00016                 INCB    73(R0)
                                1A   A1   B5 00019  1$:            TSTW    26(R1)                           ; 0655
                                     2F   12 0001C                 BNEQ    4$
                    50               61   0F 0001E                 REMQUE  (R1), DUMMY                      ; 0656
                                04   AC   DD 00021                 PUSHL   FCB                              ; 0657
           0000G    CF               01   FB 00024                 CALLS   #1, DEL_EXTFCB
                                04   AC   DD 00029                 PUSHL   FCB                              ; 0658
           0000G    CF               01   FB 0002C                 CALLS   #1, DEALLOCATE
              04    AC       0000G   CF   D1 00031                 CMPL    PRIMARY_FCB, FCB                 ; 0659
                                     04   12 00037                 BNEQ    2$
                             0000G   CF   D4 00039                 CLRL    PRIMARY_FCB
              04    AC       0000G   CF   D1 0003D  2$:            CMPL    DIR_FCB, FCB                     ; 0660
                                     04   12 00043                 BNEQ    3$
                             0000G   CF   D4 00045                 CLRL    DIR_FCB
                    50               01   D0 00049  3$:            MOVL    #1, R0                           ; 0663
                                          04 0004C                 RET
                                     50   D4 0004D  4$:            CLRL    R0                               ; 0665
                                          04 0004F                 RET

; Routine Size:  80 bytes,    Routine Base:  $CODE$ + 010F
```

```
1 ROUTINE DELETE_HANDLER (SIGNAL, MECHANISM) =
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is the condition handler for reading the file header.
1 !       If any errors occurr, it unwinds and returns to MARK_DELETE's caller,
1 !       causing the delete oif a bad file header to be a quiet NOP.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !       HANDLER (ARG1, ARG2)
1 !
1 ! INPUT PARAMETERS:
1 !       ARG1: address of signal array
1 !       ARG2: address of mechanism array
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       SS$_RESIGNAL or none if unwind
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
1
2 BEGIN
2
2 MAP
2       SIGNAL          : REF BBLOCK,   ! signal arg array
2       MECHANISM       : REF BBLOCK;   ! mechanism arg array
2
2 EXTERNAL
2       USER_STATUS;                    ! status to return to user
2
2
2 ! If the condition is change mode to user (error exit) cause an unwind to
2 ! return to DELETE's caller.
2 ! Otherwise, just resignal the condition.
2 !
2
2 IF .SIGNAL[CHF$L_SIG_NAME] EQL SS$_CMODUSER
2 THEN
3     BEGIN
3     USER_STATUS = .SAVE_STATUS;
3     $UNWIND ();
2     END;
```

```
;  413          0723  2
;  414          0724  2 RETURN SS$_RESIGNAL;                        ! status is irrelevant if unwinding
;  415          0725  2
;  416          0726  1 END;                                        ! end of routine DELETE_HANDLER



                                                                           .EXTRN  SYS$UNWIND

                                                 0000 00000 DELETE_HANDLER:
                                                                           .WORD   Save nothing                          ; 0666
                          50          04    AC  D0 00002                   MOVL    SIGNAL, R0                            ; 0717
                00000424  8F          04    A0  D1 00006                   CMPL    4(R0), #1060
                                            10  12 0000E                   BNEQ    1$
                    0000G CF       0000'    CF  D0 00010                   MOVL    SAVE_STATUS, USER_STATUS              ; 0720
                                            7E  7C 00017                   CLRQ    -(SP)                                 ; 0721
                00000000G 00                02  FB 00019                   CALLS   #2, SYS$UNWIND
                          50        0918    8F  3C 00020 1$:               MOVZWL  #2328, R0                             ; 0724
                                                04 00025                   RET                                           ; 0726

; Routine Size:  38 bytes,    Routine Base:  $CODE$ + 015F


;  417          0727  1
;  418          0728  1 END
;  419          0729  0 ELUDOM




;                                  PSECT SUMMARY
;
;       Name                    Bytes                       Attributes
;
;   $LOCKEDD1$                       8  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                         389  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                           Library Statistics
;
;                                          -------- Symbols --------       Pages       Processing
;       File                               Total    Loaded   Percent       Mapped      Time
;
;   _$255$DUA28:[SYSLIB]LIB.L32;1          18619       28          0       1000          00:01.9




;                                  COMMAND QUALIFIERS
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:DELETE/OBJ=OBJ$:DELETE MSRC$:DELETE/UPDATE=(ENH$:DELETE)

; Size:           389 code + 8 data bytes
; Run Time:          00:11.1
; Elapsed Time:      00:26.8
; Lines/CPU Min:      3947
; Lexemes/CPU-Min: 13608
; Memory Used:  97 pages
; Compilation Complete
```

0164 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CHKSUM
LIS

ACPCNTRL
LIS

CHKPRO
LIS

FCPDEF
B32

DEACCS
LIS

BADSCN
LIS

CLENUP
LIS

CPYNAM
LIS

CHKHDR
LIS

COMMON
LIS

CREHDR
LIS

CREWIN
LIS

ALLOCB
LIS

ACCESS
LIS

CHKDMO
LIS

DELETE
LIS

CREATE
LIS

CREFCB
LIS

0165 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY